

东方
瑞通
终身学习
Founded in 1998

JavaWeb

王东



关注微信公众号
享终身免费学习

easthome.com



• 东方瑞通专职讲师 王东

• IT从业经验8年

• 擅长技术：

- Java
- Android
- Web前端
- Python
- 数据分析与挖掘
- 大数据
- 人工智能

• 资质

- 高级软件开发工程师
- Android高级开发工程师
- 华为HCIP人工智能认证资质
- 华为HCIA大数据认证资质
- 华为HCSI授权认证讲师

课程介绍

• HTML

- 超文本标记语言(Hyper Text Markup Language)
- 介绍：<https://baike.baidu.com/item/HTML/97049?fr=aladdin>
- 作用：制作页面

• CSS

- 层叠样式表(Cascading Style Sheets)
- 介绍：<https://zhidao.baidu.com/question/447612528.html>
- 作用：渲染页面效果，提高用户体验

课程介绍

• JavaScript

- 介绍：<https://baike.baidu.com/item/javascript/321142?fr=aladdin>
- 作用：事件响应，页面交互

• JQuery库

- Javascript中的库，框架
- 介绍：<https://baike.baidu.com/item/jQuery/5385065?fr=aladdin>
- 作用：jQuery设计的宗旨是“write Less , Do More” ，即倡导写更少的代码，做更多的事情

课程介绍

- Web前端课程介绍
 - 网页及常用浏览器介绍
 - Web标准介绍
 - 开发工具介绍及安装
- HTML超文本标记语言
 - HTML超文本标记语言介绍
 - HTML中常用的标签
 - H5新增标签介绍
 - H5中新增属性介绍

课程介绍

- **CSS样式**

- CSS样式添加的三种方式
- CSS样式常用的七种选择器介绍
- CSS中的盒子模型
- CSS中块元素，内联元素，内联块元素介绍
- Margin，overflow，display，float，position，background等属性介绍
- CSS中的transition动画，transform变换和Animation动画

课程介绍

- **JavaScript**

- 原生Javascript介绍
- JavaScript引入的三种方式
- JavaScript中的变量，函数，属性等介绍
- DOM模型
- JavaScript中的流程控制及定时器的使用
- JavaScript中常用的事件监听器介绍

课程介绍

- Jquery库
 - Ajax概述
 - XMLHttpRequest使用
 - Json语法与使用
 - jQuery核心函数
 - jQuery选择器
 - jQuery基于Ajax使用

课程介绍

- Echarts可视化大图
 - Echarts官网介绍
 - Echarts资源下载
 - Echarts可视化大图的具体实现

网页介绍

• 什么是网页？

• 网页构成

- Mi.com

- 构成：文字，图片，超链接等元素，还包括音频，视频等

• 网页是如何形成

- Mi.com客户端（通过浏览器渲染后客户看到的是丰富多彩的）

- 后台代码展示

HTML

- **IDEA创建HTML文件**

- HTML文档结构介绍

- 创建（htm和html两种后缀名的方式创建网页）+快捷键Tab
 - 文档声明
 - Head：所有头部标签的容器，资源的引入
 - Body：网页主题结构

- **文本编辑及运行**

- 我的第一个页面

浏览器

• 常用浏览器介绍

- 五大浏览器厂商
- ie : windows (微软)
- Edge : win10
- Mac : safari
- 各大浏览器市场份额 : <http://tongji.baidu.com/data/browser>



IE浏览器



火狐浏览器



谷歌浏览器



Edge浏览器



Safari浏览器



Opera浏览器

WEB标准

- 为什么要有标准？

- 因为我们开发完的代码在不同厂商的浏览器上打开的页面效果可能不同，所以必须制定浏览器统一标准

- 标准制定

- W3C组织和其他标准化组织一起制定，主要包含结构，表现和行为三个方面

- 标准内容

- 结构标准：**HTML**，页面结构，内容
- 样式标准：**CSS**，页面外观，板式，颜色，大小等
- 行为标准：**Javascript**，页面事件，行为

WEB标准



开发工具安装及介绍

- **Sublime代码编辑器**

- 官网地址：<http://www.sublimetext.com/>
- 使用介绍：
 - [安装及插件包](#)
 - [快捷键汇总](#)

- **IDEA集成开发工具**

- 官网地址：<https://www.jetbrains.com/>
- 使用介绍
 - 安装及插件包
 - 快捷键汇总

HTML

- 第一个html文件

- hello.html , 开发工具

- HTML文档结构

- 文档声明
 - Html : 根标签
 - Head : 头标签 , title , meta等等
 - Body : 主体标签 : 页面主体架构

- HTML常用标签

- h标签

- 用法：自动换行功能，加粗（注意最多只有六级）
 - 作用：**重要**因为爬虫在爬取内容时会根据h标签来理解文档的结构
 - 场合：标题标签
 - 案例练习

- P标签

- 用法：主要是文档内容段落，注意转义字符：空格（ ），大于（>）小于（<）
 - 作用：段落
 - 场合：段落标签
 - 案例练习

• HTML常用标签

- br标签（单标签：不需要结尾）
 - 用法：需要换行标签的末尾添加
 - 作用：换行标签
 - 场合：换行
- 块级标签div，span
 - 用法：div表示一块内容没有具体含义，定义后可在这一块区域内添加任何标签，Span标签本身没有任何语义，不过我们可以借用span标签给某段文字加样式
 - 作用：不带有任何样式的块级标签，定义某一块区域
 - 场合：块级标签，可以定义一块内容，区域内可以放入多个标签

- HTML常用标签

- em标签，i标签，b标签，strong标签

- Em标签：为行内标签，表示语气中的强调词（加重内容）
 - i标签：为行内标签，表示专业词汇（斜体样式）
 - B标签：为行内标签，表示文档中的关键字（粗体样式）
 - Strong标签：为行内标签，表示文档中非常重要的内容（通常是用加粗的字体（相对于斜体））

- img标签

- 用法：在文档中添加图像显示
 - 作用：图片显示，src属性和alt(定义图片加载失败时显示的文字，搜索引擎会使用文字收录图片，盲人读屏软件会读取内容帮助盲人识别图片)属性

- **HTML常用标签**

- a标签

- 用法：连接到某一个网页或者地址
 - 作用：超链接
 - 场合：页面跳转或者地址链接
 - 案例：本地超链接，百度超链接
 - Title属性，target属性介绍
 - Title：鼠标悬停后提示内容
 - Target：_self(把自己替换掉，依然跳到当前页)，_blank（表示另外再打开一个页面窗口，依然去往目标页面）

HTML

• HTML常用标签

- 列表标签 (ol , li)
 - 用法：以列表形式显示
 - 作用：有序列表
 - 场合：有序列表制作（应用不是很多）
 - 案例练习
- 列表标签 (ul , li)
 - 用法：以列表形式显示
 - 作用：无序列表
 - 场合：无序列表制作（应用比较广泛）
 - 案例练习

HTML

- **HTML常用标签**

- 定义列表dl , dt , dd

- 介绍

- dl : (Definition List) 定义列表

- dt : (Definition Term) 定义术语

- dd : (Definition Description) 定义描述

- 用法：用于术语的定义

- 作用：dl表示列表的整体，dt表示术语的标题，dd表示术语的内容描述

- 案例：本地超链接，百度超链接

HTML

- HTML常用标签

- 表格标签table

- 介绍

- table : (table) 表格

- tr : (table row) 表格中的行

- td : (table data cell) 表格行中的列

- 用法：用于页面表格数据展示

- 属性：

- Border边框，align对齐，colspan合并列，rowspan合并行

- 作用：页面表格展示

- 案例：个人简历制作（网页版）

- 综合案例：布局标签div+table实现网页版个人简历

- **HTML常用标签**

- Form表单标签

- 用法：表单
 - 作用：定义整体的表单区域
 - 属性介绍：
 - Action：定义表单提交的地址
 - Method：定义表单提交的方式：get和post两种方式

- **HTML常用标签**

- input标签

- 介绍：定义通用的表单元素
 - 属性介绍：
 - Type：定义表单元素的类型
 - Value：定义表单元素的值
 - Name：定义表单元素的名称
 - 作用：页面表单提交

- **HTML常用标签**

- Type属性值

- Text：定义单行可输入文本框
 - Password：定义密码输入框
 - Radio：定义单选框
 - Checkbox：定义复选框
 - Email：定义邮箱格式输入框
 - File：定义文件上传
 - Submit：定义提交按钮
 - Reset：定义重置按钮

HTML

- **HTML常用标签**

- Textarea标签

- 用法：在表单中定义一块文本区域
 - 作用：表单中个人介绍等片区文本输入
 - 场合：表单提交中
 - 属性：rows：文本区行数，cols：文本区列数

- Select , option标签

- 用法：下拉选择框
 - 作用：主要用于下拉选择框
 - 场合：省市联动等

- **HTML5新增标签**

- H5新增标签介绍

- Datalist：类似百度搜索框输入后的下拉，定义选项列表，一般和input标签联合使用：
 - 案例：明星
 - Fieldset：将表单里面的元素分组，打包。一般和legend（标题）搭配使用
 - 案例：用户登录

- **HTML5新增标签**
 - H5新增表单属性介绍
 - 使用Fieldset案例展示
 - Email
 - Number
 - Tel
 - url
 - Search
 - Range
 - Time
 - Date
 - Month
 - Week color
 - datetime

HTML

- **HTML5新增标签**

- H5新增表单控件属性

- placeholder：占位符，隐藏提示
 - autofocus：自动获取焦点，不用用户定焦点
 - Multiple：上传多个文件
 - Autocomplete：设置表单是否自动完成，具有记忆功能，自动记录完成
 - 必须要有提交按钮
 - 必须要有name属性
 - Required：此项不能为空，必填项
 - accesskey= "s"：设置获取焦点的快捷键为alt+s

CSS

- **CSS介绍及作用**

- 介绍

- Css为层叠样式表

- 作用

- Html只负责文档的结构和内容，表现形式完全交由css负责
 - 页面样式渲染
 - 让html文档变得简洁，方便爬虫去爬去界面内容，因为爬虫只爬去html文档中的内容，不会爬去css样式

- **添加样式的三种方式**

- 方式一：直接在标签内部添加style属性后添加样式
 - 案例展示
- 方式二：直接在html文档的head头部标签内添加style标签
 - 案例展示
- 方式三：直接在html文档外部新建.css后缀的样式文件，后使用link标签在head标签内引入
 - 案例展示

- CSS常用样式介绍

- Color
 - 设置文字颜色
- Font-size
 - 设置文字大小
- Font-family
 - 设置文字的字体
- Font-style
 - 设置字体是否倾斜，normal不倾斜，italic倾斜
- Font-weight
 - 设置字体是否加粗，normal不加粗，bold加粗

- **CSS样式选择器**

- *号选择器

- 选择HTML文档中的所有标签

- 标签选择器

- 选择跟标签名相同的标签

- Id选择器

- 选择标签中id属性名
 - 特性：页面上标签的id名是唯一不能够重复的（局限性）

- Class类选择器

- Css中最常用的一种选择器，class属性是可以重复使用在不同的标签上
 - 一个class选择器可以应用到多个标签上，一个标签上也可以使用多个class选择器

- **CSS样式选择器**

- 层级选择器

- 主要应用在选择父元素下的子元素或者子元素下的子元素
 - 非常常用，一般标签选择器不单独使用，会和层级选择器一起用
 - 注意：层级选择器一般不要超过四层，不然会影响页面响应效果，影响性能

- 组选择器

- 一般多个选择器有同样的样式可以使用组选择器
 - 组选择器的每个选择器中间用，号隔开

- 伪类及伪元素选择器

- 伪类hover：一般用于连接的响应式效果，可以修改鼠标悬停样式
 - 伪元素选择器：before在标签内容前面添加内容，after在标签内容后面添加内容，注意添加的内容都是不能被选中的

CSS

- 盒子模型



- 盒子模型

- 盒子的基本设置
 - Width宽，height高
- Border属性设置
 - Border-width：边框宽，border-style：边框样式，border-color：颜色
 - Border综合写法，三个属性一同写入
- Padding属性设置
 - Padding-top上内边距，right，bottom，left
 - 综合写法：顺时针设置属性，三个值（上，左右，下）两个值（上下，左右）
- Margin属性设置
 - Margin顺时针写法
 - 注意：margin属性独有的，可以使元素水平居中

CSS

Margin属性扩展

- Auto
 - 设置后可以实现元素水平方向居中，注意auto属性只能用于水平方向
 - 案例展示
- Margin设置负值
 - 盒子实现位移（反方向），注意body标签在浏览器上本身有8px边距
 - 案例展示
- Margin边距合并
 - 同一方向的两个元素都添加margin的话只会执行那个数值大的（只会执行一个）
 - 案例展示
- Margin顶部塌陷
 - 出现在另个盒子嵌套使用时，内部盒子添加margin-top属性时会自动传递给外部盒子，而内部盒子的margin-top不起作用

CSS

- **盒子模型：块元素，内联元素，内联块元素**

- 块元素

- 概念：块元素也可以称为行元素，整个元素定义后占用一行
- 常见块元素标签：div，p，ul，li，h1-h6，dl，dt，dd等
- 特性：
 - 支持全部的样式
 - 如果没有设置宽度，默认宽度为父级宽度的100%
 - 独自占据一行，即使设置了宽度

CSS

- **盒子模型：块元素，内联元素，内联块元素**

- 内联元素

- 概念：内联元素也可以称为行内元素，元素不会独自占据一行
- 常见内联元素标签：a，span，em，b，strong，i等
- 特性：
 - 支持部分样式（不支持宽高，margin上下）
 - 宽高由内容决定
 - 盒子并在一起
 - 元素换行，盒子之间会产生间隙
 - 子元素是内联元素，父元素可以使用text-align属性设置子元素水平对齐方式

- **盒子模型：块元素，内联元素，内联块元素**

- **内联块元素**

- 概念：内联块元素也可以称为行内块元素，元素独自占据一行
- 内联块元素是新增的元素类型，没有特别划分，我们可以通过设置来把块元素和内联元素转化为内联块元素
- 好处：不仅解决了块元素独占一行的问题，也解决了内联元素不能设置宽高的问题
- 特性：
 - 支持全部样式
 - 不设置宽高时，宽高由内容决定
 - 盒子并在一起
 - 元素换行，盒子之间会产生间隙
 - 子元素是内联元素，父元素可以使用text-align属性设置子元素水平对齐方式

- **盒子模型**

- 解决元素换行产生的间距
 - 去掉内联元素之间换行
 - 将内联元素父级font-size设置为0，内联元素自身再设置font-size
 - 案例练习
 - 新闻菜单div，a

- **盒子模型**

- display

- 用法：用来设置元素的类型及隐藏及隐藏
 - 常用值
 - None：元素隐藏，且不占用位置
 - Block：元素设置为块元素类型
 - Inline：元素设置为内联元素类型
 - Inline-block：元素设置为内联块元素类型
 - 案例练习
 - UI，li新闻导航条练习

- 盒子模型

- Float浮动

- 只能设置元素左右浮动

- 特性：

- Float : left , float : right
 - 相邻浮动的块元素可以并在一行，超出父级宽度就换行
 - 浮动可以让行内元素和块元素自动转化为行内块元素（并且不会有行内块元素间隙问题）
 - 浮动元素后面没有浮动的元素，那么没有浮动的元素中的文字会避开浮动元素的位置，形成文字绕图
 - 父元素如果没有设置尺寸，那么父元素内整体浮动的子元素无法撑开父元素，这时父元素需要清除浮动
 - 浮动元素之间没有垂直margin的合并

• 盒子模型

• 清楚浮动的三种方式

- 方式一：父元素添加属性：overflow ; hidden
- 方式二：在最后一个子元素后面添加一个div，给他样式属性为clear : both（清除两端）不推荐使用
- 方式三：使用伪元素after，clearfix类添加到父级元素即可，具体代码：
 - Content : " " , display:table , clear:both
 - 最终既要解决margin-top顶部塌陷，又要解决清除浮动，代码经常这样写
 - Zoom : 1解决的是ie浏览器不识别after属性的问题，ie现在叫做非标准浏览器

```
.clearfix:before,.clearfix:after{  
    content:"";  
    display:table;  
}  
.clearfix:after{  
    clear:both;  
}  
.clearfix{  
    zoom:1;  
}
```

- **盒子模型**

- Position定位

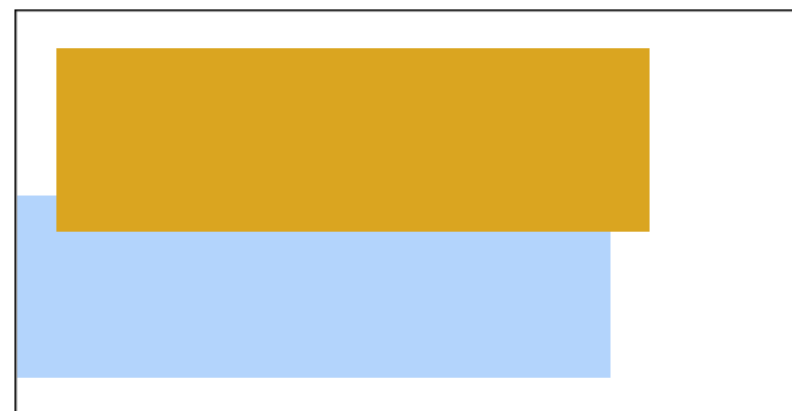
- 文档流概念（为什么使用定位？）：文档流就是盒子在文档中排列的顺序，默认是从左向右，从上向下排列的，如果要改变盒子的位置，这个时候就涉及到定位了
- 定位方式
 - 相对定位position : relative
 - 绝对定位position : absolute
 - 固定定位position : fixed

CSS

• 盒子模型

• Relative相对定位

- 原理：生成相对定位的元素，元素所占据的文档流的位置保留，元素本身相对于自身的原来位置进行偏移
- 定位一般和float属性联合使用

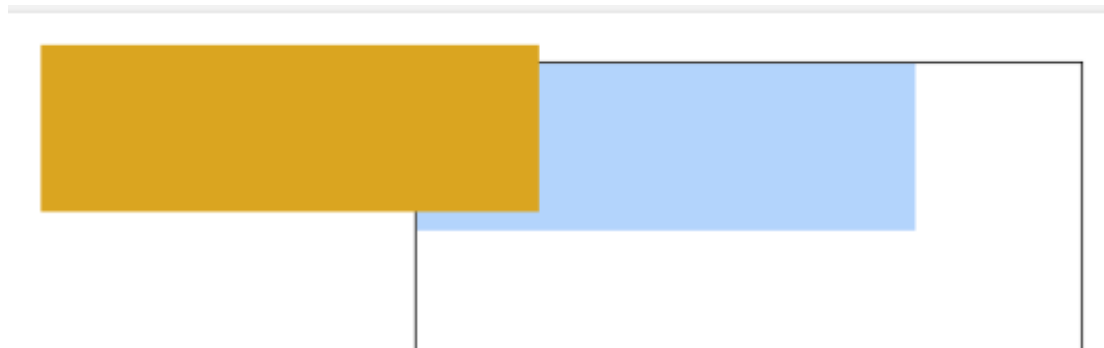


- 盒子模型

- Absolute绝对定位

- 原理：生成绝对定位的元素，元素脱离文档流，不占用文档流的位置，相当于漂浮在文档流的上方，定位是相对于已经设置了定位属性的父级元素定位，如果父级没有设定，那么就相对与body定位

- 案例练习：微信头像



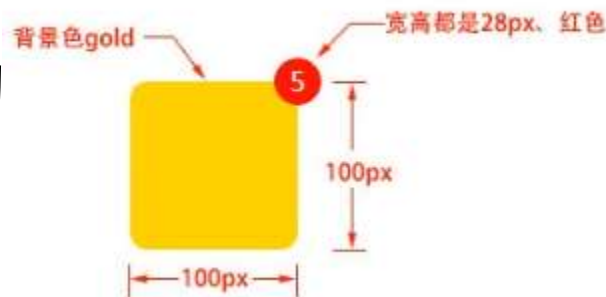
CSS

• 盒子模型

• Fixed固定定位

- 原理：生成固定定位的元素，元素脱离文档流，不占用文档流的位置，相当于漂浮在文档流的上方，定位是相对于浏览器进行定位

• 案例练习



CSS

- 盒子模型

- Static默认定位

- 原理：static默认值，没有定位，元素正常出现在文档流位置，相当于取消定位属性
 - 注意：固定定位和绝对定位的块元素和行内元素都会转换为行内块元素

CSS盒子模型

- Background属性
 - 作用：使用比较频繁，主要用来给元素设置背景颜色和背景图片，复合属性
 - 属性：
 - Background-color：设置背景颜色
 - Background-image：设置背景图地址url
 - Background-repeat：设置背景图片重复方式
 - Background-position：设置背景图片的位置
 - Background-attachment:设置背景图片是否随着页面滚动而滚动
 - Background：复合设置

CSS

• 盒子模型

- CSS圆角radius和透明度rgba
 - radius属性：
 - Border-top-left-radius：设置某一个角的圆角
 - Border-radius：顺时针同时分别设置四个角的圆角
 - Border-radius：设计四个圆角相同
 - Rgba透明度
 - opacity: 0.6
 - 复合写法：background-color: rgba(0,0,0,0.4);

CSS

• CSS3变换效果

• Transform变换（和动画联合使用）

- 和动画联合使用，就是给某个transform变换添加动画

CSS3 transform变换

- 1、translate(x,y) 设置盒子位移
- 2、scale(x,y) 设置盒子缩放
- 3、rotate(deg) 设置盒子旋转
- 4、skew(x-angle,y-angle) 设置盒子斜切
- 5、perspective 设置透视距离
- 6、transform-style flat | preserve-3d 设置盒子是否按3d空间显示
- 7、translateX、translateY、translateZ 设置三维移动
- 8、rotateX、rotateY、rotateZ 设置三维旋转
- 9、scaleX、scaleY、scaleZ 设置三维缩放
- 10、transform-origin 设置变形的中心点
- 11、backface-visibility 设置盒子背面是否可见

CSS

• CSS3动画animation

- animation动画
 - 特点：不用触发，自己直接执行
- 案例：
 - 心跳
 - 大风车转动
 - 页面loading动画

CSS3 animation动画

- 1、@keyframes 定义**关键帧动画**
- 2、animation-name 动画名称
- 3、animation-duration 动画时间
- 4、animation-timing-function 动画曲线 linear(匀速)|ease(缓冲)|steps(步数)
- 5、animation-delay 动画延迟
- 6、animation-iteration-count 动画播放次数 n|infinite
- 7、animation-direction 动画结束后是否反向还原 normal|Alternate
- 8、animation-play-state 动画状态 paused(停止)|running(运动)
- 9、animation-fill-mode 动画前后的状态 none(缺省)|forwards(结束时停留在最后一帧)|backwards(开始时停留在定义的开始帧)|both(前后都应用)
- 10、animation: name duration timing-function delay iteration-count direction;同时设置多个属性

CSS

• CSS权重

- Ccss样式权重计算

- 案例：

- 鼠标悬停时文字划入说明效果

1、!important，加在样式属性值后，权重值为 10000

2、内联样式，如：style=""，权重值为1000

3、ID选择器，如：#content，权重值为100

4、类，伪类和属性选择器，如：content、:hover 权重值为10

5、标签选择器和伪元素选择器，如：div、p、:before 权重值为1

6、通用选择器（*）、子选择器（>）、相邻选择器（+）、同胞选择器（~）、权重值为0

1.5.1 简介

- JavaScript 是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言。它的解释器被称为 JavaScript 引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在 HTML 网页上使用，用来给 HTML 网页增加动态功能。
 - 可以实现和 html 的交互
 - 可以实现动态的样式修改 (css)
 - 实现对表单的校验
 - 动态对事件进行响应

JavaScript

- **原生JavaScript**

- 引入原生js的三种方式（类似于css的引入）
 - 方式一：在标签内添加js代码（行间事件），主要用于事件
 - 方式二：html页面前途script标签，引入type属性
 - 方式三：外部引入js文件，src属性引入

1.5.2 语法

- 引入方式
 - 外联：`<script type="text/javascript" src="js/myjs.js" ></script>`
 - 内嵌：`<script></script>`
- 输出数据
 - 使用 `window.alert()` 弹出警告框。
 - 使用 `document.write()` 方法将内容写到 HTML 文档中。
 - 使用 `innerHTML` 写入到 HTML 元素。
 - 使用 `console.log()` 写入到浏览器的控制台。

1.5.2 语法

- 注释
 - 单行注释以 // 开头
 - 多行注释以 /* 开始，以 */ 结尾
- 输出数据
 - 使用 window.alert() 弹出警告框。
 - 使用 document.write() 方法将内容写到 HTML 文档中。
 - 使用 innerHTML 写入到 HTML 元素。
 - 使用 console.log() 写入到浏览器的控制台。

1.5.3 变量和数据类型

- 数据类型：
 - 值类型(基本类型)：
 - 字符串 (String)、数字(Number)、布尔(Boolean)、对空 (Null)、未定义 (Undefined)、Symbol。
 - 引用数据类型：
 - 对象(Object)、数组(Array)、函数(Function)
- 变量
 - 定义变量：var 变量名
 - 局部：定义在函数内部，所以只能在函数内部访问它
 - 全局：在函数外声明的变量是全局变量，网页上的所有脚本和函数 都能访问它

```
function abc(){
}
var a = "aaa";
var b = 3.14;
var c = true;
var d = ["a", "b", "c"];
var e = null;
var f = {"name": "zs", "age": 13};
var g
console.log(typeof a)
console.log(typeof b)
console.log(typeof c)
console.log(d)
console.log(e)
console.log(f)
console.log(typeof g)
```

```
string
number
boolean
▼ (3) ["a", "b", "c"] ⓘ
  0: "a"
  1: "b"
  2: "c"
  length: 3
  ▶ __proto__: Array(0)
null
▼ {name: "zs", age: 13} ⓘ
  name: "zs"
  age: 13
  ▶ __proto__: Object
undefined
```

1.5.4 运算符

- 算术运算符
 - +、-、*、/、%、++、--
- 赋值运算符
- +=、-=、*=、/=、%=
- 比较运算符:
 - ==（比较值）、===（比较值和类型）、>=、<=、>、<、!=、!==
- 逻辑运算符：&&、||、!
- 三目运算符：a > b ? a : b

2.5.5 流程控制

- if 语句
- switch 语句
- 循环
 - while
 - do...while
 - for
 - for ... in (遍历对象的属性)

```
switch(n)
{
    case 1:
        执行代码块 1
        break;
    case 2:
        执行代码块 2
        break;
    default:
        与 case 1 和 case 2 不同时执行的代码
}
```

```
for (语句 1; 语句 2; 语句 3)
{
    被执行的代码块
}
```

语句 1 (代码块) 开始前执行

语句 2 定义运行循环 (代码块) 的条件

语句 3 在循环 (代码块) 已被执行之后执行

```
var person={fname:"John",lname:"Doe",age:25};
```

```
for (x in person) // x 为属性名
{
    txt=txt + person[x];
}
```

1.5.6 函数

- 函数：定义在 script 标签内的可被重复执行的代码块

- 无参函数
- 带参函数
- 有返回值函数
- 匿名函数（事件触发）
- 内置函数：
 - alert()、
 - console.log()、
 - confirm()、
 - parseInt()
 - setInterval() 等等

```
function test1() {
    alert("国庆快乐")
}
```

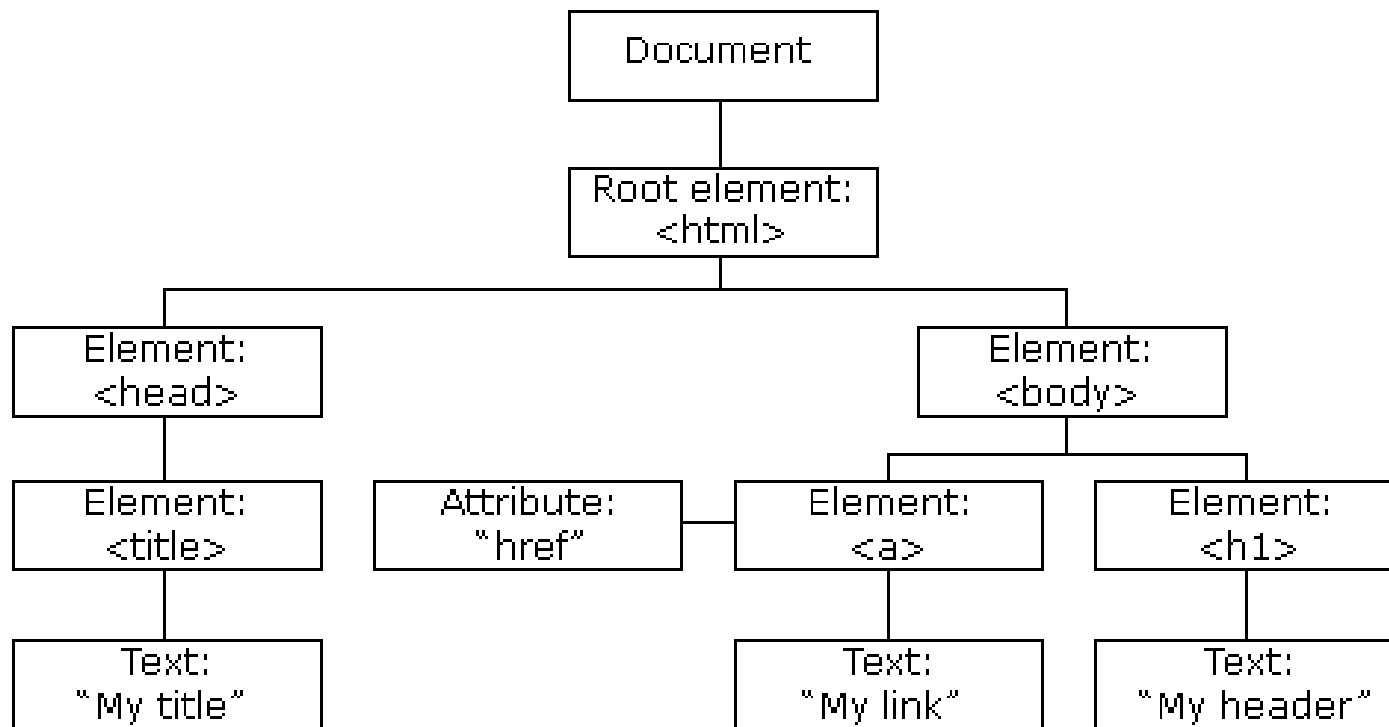
```
function add1(a, b) {
    alert(eval(a + b))
}
```

```
function add2(a, b) {
    return a + b
}
```

```
document.onkeydown = function(e) {
    if(e.keyCode==13){
        alert("你按了回车键！")
    }
}
```

1.5.7 基本dom 操作

- 当网页被加载时，浏览器会创建页面的文档对象模型（Document Object Model），JavaScript 通过 DOM，可访问 HTML 文档的所有元素。HTML DOM 模型被构造为对象的树：



1.5.7 基本dom 操作

- WEB 页面的 HTML 文档，document就是根节点，其它的子对象就是子结点
 - document.getElementById : (id的值); 获取到元素是唯一的
 - document.getElementsByTagName : (标签名) ; 获取到元素是一个数组
 - document.getElementsByName : (name的值); 获取到元素不是唯一
 - document.getElementsByClassName : (class的值); 获取到元素不是唯一

1.5.8 DOM 中常用的事件函数

- //失去焦点函数
- //聚焦函数
- //当内容发生改变时进行该函数的调用
- //当元素被点击的时候进行调用
- //当一个页面或者是一幅图像被加载完成的时候调用
- onblur();
- onfocus();
- onchange();
- onclick();
- onload();
- onmousedown(); //当鼠标点击是触发的事件
- onmouseup(); //当鼠标松开时触发的事件
- onmouseover(); //鼠标移入某元素是触发的事件

1.5.9 JavaScript—基本 dom 操作

- 案例：简易计算器
- 第3个输入框获得鼠标焦点时，计算出前2个输入框之和

311 + 4 = 315

```
<input id="add1" type="text" />+
<input id="add2" type="text" />=
<input id="sum" type="text" onfocus="add1()" />
<script type="text/javascript">
    function add1() {
        var oAdd1 = document.getElementById("add1");
        //获取第一个输入框的值
        var a = oAdd1.value;

        var oAdd2 = document.getElementById("add2");
        //获取第二个输入框的值
        var b = oAdd2.value;

        var oSum = document.getElementById("sum");
        //设置第三个输入框的值
        oSum.value = parseInt(a) + parseInt(b);
    }
</script>
```

1.6.1 jQuery 简介


- jQuery 是一个JavaScript函数库
- jQuery是一个轻量级的"写的少，做的多"的JavaScript库
- jQuery 能帮我们干这些事情：
 - 消除浏览器差异：不需要自己写冗长的代码来针对不同的浏览器来绑定事件，编写 ajax 等代码；
 - 简洁的操作 DOM 的方法：写 `$("#test")`肯定比 `document.getElementById('test')`来得简 洁；
 - 轻松实现动画、修改 CSS 等各种操作。

1.6.2 下载及引用

- 从jquery.com 下载jQuery 库

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to you.

[Download the compressed, production jQuery 3.4.1](#) 

[Download the uncompressed, development jQuery 3.4.1](#)

[Download the map file for jQuery 3.4.1](#)

- 我们在使用之前得在 <script> 标签中引入 jQuery 文件：

```
<script src="js/jquery-3.4.1.min.js"></script>
```

1.6.3 jQuery的简单使用

- 基础语法：\$(selector).action()
 - 美元符号定义 jQuery
 - 选择符 (selector) "查询"和"查找" HTML 元素
 - jQuery 的action() 执行对元素的操作

<code>\$(this).hide()</code>	- 隐藏当前元素
<code>\$("p").hide()</code>	- 隐藏所有 <code><p></code> 元素
<code>\$("p.test").hide()</code>	- 隐藏所有 <code>class="test"</code> 的 <code><p></code> 元素
<code>\$("#test").hide()</code>	- 隐藏所有 <code>id="test"</code> 的元素

1.6.3 jQuery的简单使用

- jQuery 选择器，\$(selector)
 - jQuery 中所有选择器都以美元符号开头：\$()
 - 选择器基于元素的 id、类、类型、属性、属性值等查找 HTML 元素。
 - 它基于已经存在的 CSS 选择器，除此之外，它还有一些自定义的选择器。

语法	描述
<code>\$("*")</code>	选取所有元素
<code>\$(this)</code>	选取当前 HTML 元素
<code>\$("#p.intro")</code>	选取 class 为 intro 的 <p> 元素
<code>\$("#p:first")</code>	选取第一个 <p> 元素
<code>\$("#ul li:first")</code>	选取第一个 元素的第一个 元素
<code>\$("#ul li:first-child")</code>	选取每个 元素的第一个 元素
<code>\$("#[href]")</code>	选取带有 href 属性的元素
<code>\$("#a[target='_blank']")</code>	选取所有 target 属性值等于 "_blank" 的 <a> 元素
<code>\$("#a[target!='_blank']")</code>	选取所有 target 属性值不等于 "_blank" 的 <a> 元素
<code>\$("#:button")</code>	选取所有 type="button" 的 <input> 元素 和 <button> 元素
<code>\$("#tr:even")</code>	选取偶数位置的 <tr> 元素
<code>\$("#tr:odd")</code>	选取奇数位置的 <tr> 元素

1.6.3 jQuery的简单使用

- 常见 DOM 事件

鼠标事件	键盘事件	表单事件	文档/窗口事件
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

- jQuery事件监听：
 - \$(选择器).事件名（func）

```

$("p").click(function(){
    // 动作触发后执行的代码!!
});

```


1.6.3 jQuery的简单使用

- 获取与设置内容和属性

简单实用的用于 DOM 操作的jQuery 方法：

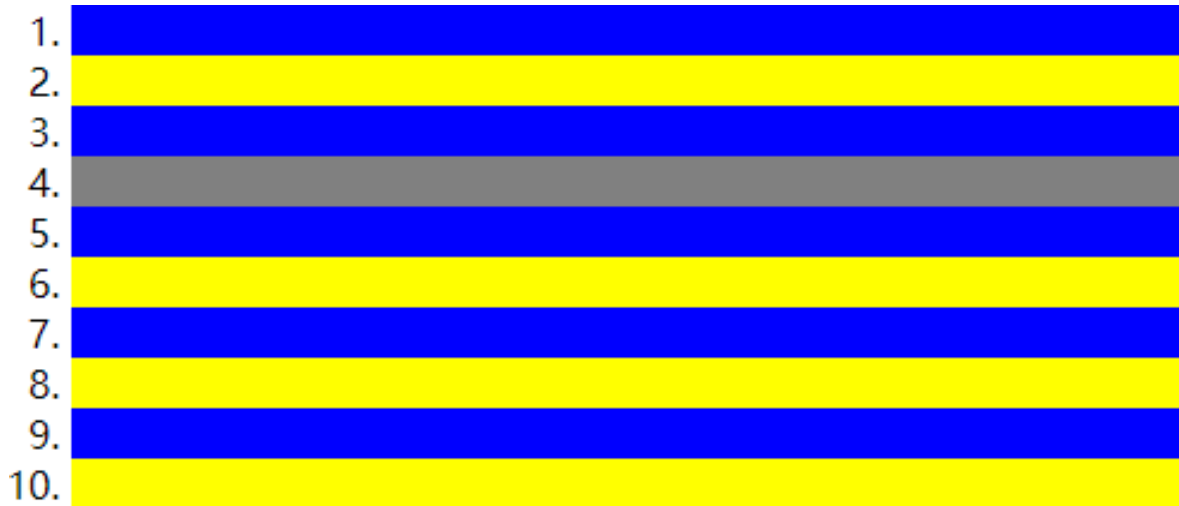
- text() - 设置或返回所选元素的文本内容
- html() - 设置或返回所选元素的内容（包括 HTML 标记）
- val() - 设置或返回表单字段的值
- css() - 方法为被选元素设置或返回一个或多个样式属性。
- attr() - 设置或返回所选元素的属性

```
$("#button").click(function(){
    // 修改单个属性, 逗号
    $("#summit").attr({"class","a"});
});
```

```
$("#button").click(function(){
    $("#summit").attr({
        "name" : "zhangsan",
        "class" : "a"
    });
});
```

1.6.3 jQuery的简单使用

- 案例
- jQuery 实现隔行变色功能



```
<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $(function(){
        $('body').append("<ol></ol>")
        for (var i = 0; i < 10; i++) {
            $('ol').append("<li></li>")
        }
        $('li:odd').css("background","yellow")
        $('li:even').css("background","blue")
        var bgcolor
        $("li").mouseover(function(){
            bgcolor = $(this).css("background")
            $(this).css("background","gray")
        })
        $("li").mouseout(function(){
            $(this).css("background",bgcolor)
        })
    })
</script>
```

1.6.4 AJAX 发送异步请求

AJAX 是 **Asynchronous JavaScript and XML** 的简称，通过 ajax 向服务器发送 HTTP 请求，接收服务器返回的 JSON 数据，然后使用 JavaScript 修改网页的局部来实现更新和提交。现在绝大多数 Web 应用都在使用 Ajax。

1.6.4 AJAX 发送异步请求

- 原生JS Ajax 请求语法格式：

```
var xhr = new XMLHttpRequest();
xhr.open('post', '/ajax' );
//post请求一定要添加请求头才行否则会报错
xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");
//发送请求
// xhr.send('name=张三&age=18');
xhr.send({"name":"张三","age":18});
xhr.onreadystatechange = function () {
    // 这步为判断服务器是否正确响应
    if (xhr.readyState == 4 && xhr.status == 200) {
        console.log(xhr.responseText);
    }
};
```

1.6.4 AJAX 发送异步请求

- jQuery Ajax 请求语法格式：

```
$.ajax({
    type:"post",    //请求类型
    url:"/目标路由", //请求地址
    data:{"k1":"v1","k2":"v2"}, //数据
    datatype:"json", //返回的数据类型
    success:function (datas) {
        //请求成功的回调函数，datas 是返回的数据
    },
    error:function () {
        //请求失败时执行
    }
})
```

1.7 Echarts—简介

ECharts，缩写来自 Enterprise Charts，商业级数据图表，是百度的一个开源的数据可视化工具，一个纯 Javascript 的图表库，能够在 PC 端和移动设备上流畅运行，兼容当前绝大部分浏览器（IE6/7/8/9/10/11，chrome，firefox，Safari 等），底层依赖轻量级的 Canvas 库 ZRender，ECharts 提供直观，生动，可交互，可高度个性化定制的数据可视化图表。创新的拖拽重计算、数据视图、值域漫游等特性大大增强了用户体验，赋予了用户对数据进行挖掘、整合的能力。

- 目前已成为 Apache 的孵化项目
- 官网网站：<https://echarts.apache.org/zh/index.html>

1.7 Echarts—简介

可以通过以下几种方式获取 Echarts：

- ① 从官网下载界面选择你需要的版本下载，根据开发者功能和体积上的需求，我们提供了不同打包的下载，如果你在体积上没有要求，可以直接下载完整版本。开发环境建议下载源代码版本，包含了常见的错误提示和警告。
- ② 在 ECharts 的 GitHub 上下载最新的release 版本，解压出来的文件夹里的 dist 目录里可以找到最新版本的 echarts 库。
- ③ 通过 npm 获取 echarts，`npm install echarts --save`
- ④ cdn 引入，你可以在 cdnjs，npmcdn 或者国内的 bootcdn 上找到 ECharts 的最新版

1.7 Echarts—简介

- 第一步，在<script>标签中引入 Echarts 文件及 jQuery

```
<script src="../../static/js/echarts.min.js"></script>
<script src="../../static/js/jquery-1.11.1.min.js"></script>
```

- 第二步，准备一个放图表的容器

```
<!-- 为ECharts准备一个具备大小（宽高）的DOM容器1-->
<div id='pane' style="height: 500px;"></div>
```

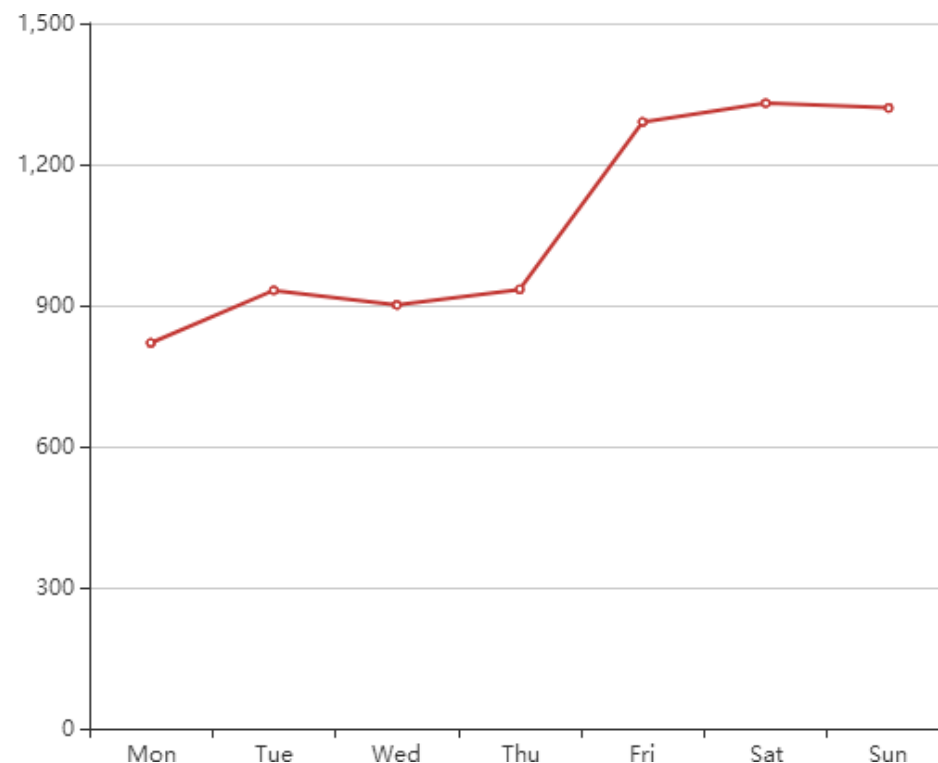

1.7 Echarts—简介

- 第三步，设置参数，初始化图表

```
var option = {
  xAxis: {
    type: 'category',
    data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
  },
  yAxis: {
    type: 'value'
  },
  series: [{
    data: [820, 932, 901, 934, 1290, 1330, 1320],
    type: 'line'
  }]
};

// 基于准备好的dom, 初始化echarts图表
var my_chart = echarts.init(document.getElementById('pane'));

//为echarts对象加载数据
my_chart.setOption(option);
```

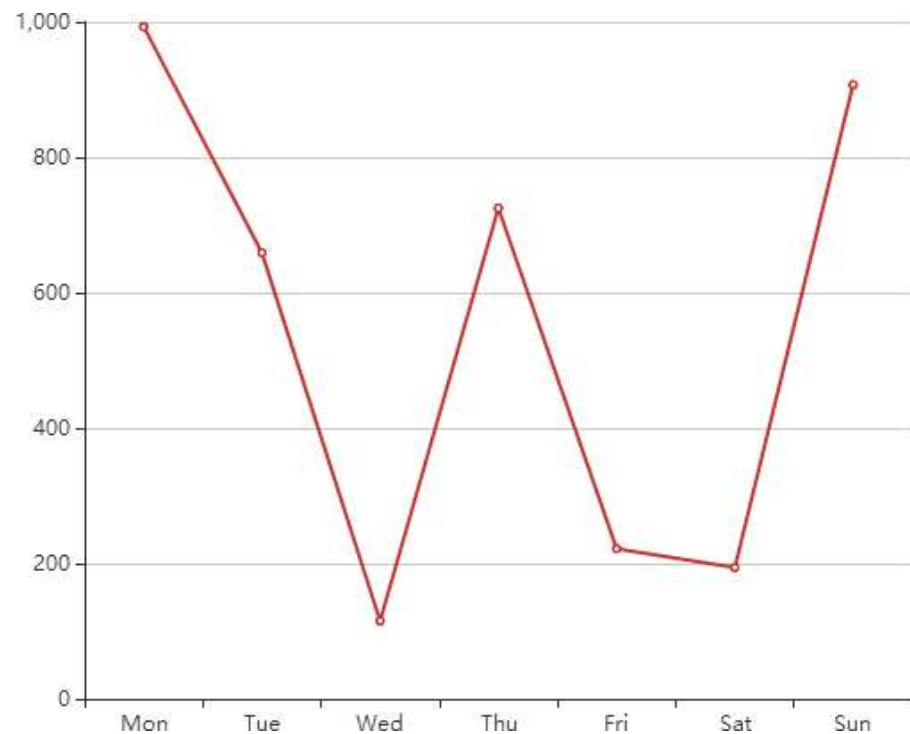


1.7 Echarts—简介

- 第四步，更改对应option里面data的值，让图表动起来

```
$("#change").click(function () {
    // $.ajax 从后台获取数据

    var d = new Array(7)
    for (var i = 0; i < 7 ; i++) {
        d[i] = Math.floor(Math.random()*1000+1);
    }
    // console.log(d)
    option.series[0].data=d
    my_chart.setOption(option)
});
```



东方
瑞通
终身学习
Founded in 1998

THANKS !



关注微信公众号
享终身免费学习

easthome.com